Research - Cryptographic Notarization Across Independent Ledgers: Cross-Chain Anchoring Protocols

Alpasan, Lois-Kleinner

2026

Abstract

Organizations operating multiple AI systems generate independent cryptographic ledgers that may need mutual verification, cross-referencing, or consolidated audit for enterprise-wide compliance reporting. Cross-chain notarization provides cryptographic evidence that a ledger's state is acknowledged by another independent ledger, enabling distributed audit verification without central coordination. This paper presents the design and analysis of the AIOSS cross-chain notarization protocol, which anchors the hash chain head of one ledger into another by inserting a notarization entry containing the cross-chain proof. We define three notarization modes: unilateral (ledger A notarizes ledger B's state), bilateral (mutual notarization between A and B), and supervised (third-party notarizer with independent proof). The notarization proof comprises a Merkle inclusion proof of the source ledger's state proof within a notarization ledger entry, enabling verification by any party holding both ledger files. We analyze the security of cross-chain anchoring under the common prefix assumption, proving that notarization preserves the integrity of both ledgers. Performance benchmarks demonstrate that notarization completes in under 200 milliseconds for ledgers of up to 1 million entries. We further evaluate the notarization merge operation, which produces a unified ledger from multiple notarized ledgers with cross-reference integrity. The protocol supports regulatory requirements for multi-sy

Cryptographic Notarization Across Independent Ledgers: Cross-Chain Anchoring Protocols

Document ID: AIOSS-RES-009-001 **Version:** 1.0.0 **Date:** 2026-06-20 **Classification:** Academic Research

---

Abstract

Organizations operating multiple AI systems generate independent cryptographic ledgers that may need mutual verification, cross-referencing, or consolidated audit for enterprise-wide compliance reporting. Cross-chain notarization provides cryptographic evidence that a ledger's state is acknowledged by another independent ledger, enabling distributed audit verification without central coordination. This paper presents the design and analysis of the AIOSS cross-chain notarization protocol, which anchors the hash chain head of one ledger into another by inserting a notarization entry containing the cross-chain proof. We define three notarization modes: unilateral (ledger A notarizes ledger B's state), bilateral (mutual notarization between A and B), and supervised (third-party notarizer with independent proof). The notarization proof comprises a Merkle inclusion proof

of the source ledger's state proof within a notarization ledger entry, enabling verification by any party holding both ledger files. We analyze the security of cross-chain anchoring under the common prefix assumption, proving that notarization preserves the integrity of both ledgers. Performance benchmarks demonstrate that notarization completes in under 200 milliseconds for ledgers of up to 1 million entries. We further evaluate the notarization merge operation, which produces a unified ledger from multiple notarized ledgers with cross-reference integrity. The protocol supports regulatory requirements for multi-system audit consolidation under SOC2 reporting and GDPR Article 30 record-of-processing activities.

---

1. Introduction

Enterprise AI deployments rarely operate in isolation. A single organization may run dozens of AI models across multiple departments, geographic regions, and regulatory regimes. Each AI system generates an independent audit ledger under the AIOSS format. Enterprise-wide compliance requires the ability to verify relationships between these ledgers: proving that a model's training data was logged before its deployment, that a compliance review preceded a production release, or that a cross-border data transfer has documented authorization.

Cross-chain notarization addresses this requirement by enabling one ledger to cryptographically anchor another ledger's state. The anchoring creates a verifiable dependency between ledgers: an auditor can verify that ledger B's state at time T is acknowledged in ledger A's entry at time T'. This enables construction of a directed acyclic graph (DAG) of ledger dependencies that mirrors organizational audit relationships.

The AIOSS notarization protocol defines three operation types: notarization (anchoring one ledger's state into another), merge (combining multiple notarized ledgers into a single unified ledger), and verification (validating cross-chain references). This paper formalizes these operations and analyzes their security and performance characteristics.

2. Literature Review

2.1 Cross-Chain Protocols in Blockchain Systems

Cross-chain communication has been extensively studied in the blockchain literature. Sidechains (Back et al., 2014) enable asset transfer between Bitcoin and alternative blockchains through SPV (Simplified Payment Verification) proofs. Zamyatin et al. (2019) provided a comprehensive survey of cross-chain interoperability protocols, identifying notarization, hash time-locked contracts (HTLCs), and relay chains as primary approaches.

The Inter-Blockchain Communication (IBC) protocol (Goes, 2020) for Cosmos provides standardized cross-chain packet verification using light clients and Merkle proofs. IBC's security model assumes trust in the source chain's consensus, analogous to AIOSS's trust in the source ledger's hash chain integrity.

Polkadot's XCMP (Cross-Chain Message Passing, Wood, 2016) uses a relay chain architecture where each parachain's state is committed to the relay chain. This is similar to cross-chain notarization but requires a centralized relay.

2.2 Inter-Ledger Notarization

The concept of notarization between independent logs was explored by Kelsey and Schneier (1999) in the context of secure audit logging across administrative domains. Crosby and Wallach (2009) proposed inter-log linking for tamper-evident storage across distributed systems.

Certificate Transparency's inclusion of "precertificates" (Langley et al., 2013) allows one log to reference another, enabling the Signed Certificate Timestamp (SCT) ecosystem. Chained CT logs provide a practical deployment of cross-log notarization at internet scale.

Tomescu et al. (2020) introduced the concept of "cross-linking" in transparency logs, where multiple logs mutually reference each other to prevent log forking attacks. Papadopoulos et al. (2021) formalized "accountable transparency logs" with cross-referencing between independent log operators.

3. Technical Analysis

3.1 AIOSS Notarization Protocol

Let (L_A) and (L_B) be two independent AIOSS ledgers. Notarization of (L_B)'s state into (L_A) proceeds as follows:

```
Algorithm 1: Cross-Chain Notarization
Input: Source ledger L_B, target ledger L_A,
       notarization key k_notary
Output: Notarized target ledger L_A' with n'

1:  _B ← L_B.state_proof              // Current state proof of L_B
2: n_entry ← CreateNotarizationEntry(
3:   source_id = L_B.chain_id,
4:   source_proof =  _B,
5:   source_head = L_B.chain_head,
6:   notary_key = k_notary
7: )
8: L_A' ← L_A.Append(n_entry)
9: return L_A'
```

The notarization entry has a dedicated entry type and includes:

```
notarization_entry = {
  entry_type: 5,                          // NOTARIZATION
  source_chain_id: [32 bytes],            // SHA3-256 of chain name
  source_state_proof: [variable],         // Full Ed25519 state proof
  source_chain_head: [32 bytes],          // Hash chain head of source
  notarization_timestamp: [8 bytes],      // Unix nanoseconds
  notarization_signature: [64 bytes],     // Ed25519 sig over above
  cross_hash: [32 bytes],                 // SHA3-256(source_head || target_head)
}
```

3.2 Notarization Modes

Unilateral Notarization (Mode 0): One ledger references another. The target ledger contains a notarization entry referencing the source ledger's state. Verification requires access to both ledgers.

```
L_A: ... n_x, n_{x+1}, ... (no reference to L_B)
L_B: ... n_notarize(L_A) ...
```

Bilateral Notarization (Mode 1): Both ledgers mutually reference each other's state, creating a cryptographic handshake:

```
L_A: ... n_notarize(L_B) ...
L_B: ... n_notarize(L_A) ... (after L_A's notarization)
```

Bilateral notarization prevents "forking" where one ledger could have different state at the time of claimed notarization. The mutual dependency ensures either both references are valid or neither is.

Supervised Notarization (Mode 2): A third-party notary ledger (L_N) independently attests to the state of both (L_A) and (L_B):

```
L_N: ... n_notarize(L_A), n_notarize(L_B) ...
```

This mode is useful for regulatory audits where a compliance authority maintains an independent notarization ledger.

3.3 Verification Algorithm

```
Algorithm 2: Cross-Chain Verification
Input: Target ledger L_A, source ledger L_B,
       notarization entry index i in L_A
Output: Boolean (is the cross-chain reference valid?)

1: n_entry ← L_A[i]                      // Read notarization entry
2: if n_entry.entry_type  NOTARIZATION then
3:   return false
4: end if
5:
6: // Verify the notarization entry's hash chain integrity
7: if VerifyHashChainEntry(L_A, i) = false then
8:   return false
9: end if
10:
11: // Verify the notarization signature
12: if VerifySignature(n_entry.notarization_signature,
13:                    n_entry.source_chain_head ||
14:                    n_entry.notarization_timestamp,
15:                    notary_key) = false then
16:   return false
17: end if
18:
19: // Verify the source ledger state at time of notarization
20: source_proof ← n_entry.source_state_proof
21: if VerifyStateProof(L_B, source_proof) = false then
22:   return false
23: end if
```

```
24:
25: // Verify the cross_hash integrity
26: cross_hash ← SHA3-256(n_entry.source_chain_head || L_A[i].hash)
27: if cross_hash   n_entry.cross_hash then
28:   return false
29: end if
30:
31: return true
```

3.4 Ledger Merge

The merge operation combines multiple notarized ledgers into a single unified ledger:

```
Algorithm 3: Merge Notarized Ledgers
Input: Ledger list L[0..n-1], merge policy P
Output: Unified ledger U

1: U ← CreateEmptyLedger(P.config)
2: for each ledger L_i in topological_order(L) do
3:   for each entry e in L_i do
4:     if e.entry_type = NOTARIZATION then
5:       // Resolve cross-reference: replace with pointer to merged entry
6:       u_entry ← CreateMergeEntry(e)
7:     else
8:       u_entry ← CopyEntry(e)
9:     end if
10:    U.Append(u_entry)
11:  end for
12:  U.Append(MergeBoundary(L_i.chain_id))
13: end for
14: return U
```

The merge operation preserves all integrity proofs: each entry retains its original hash chain linkage, and merge boundary markers enable selective re-verification of individual source ledgers.

3.5 Security Analysis

Theorem 1 (Cross-Chain Integrity Preservation): If ledgers (L_A) and (L_B) satisfy hash chain integrity individually, and (L_A) contains a valid notarization of (L_B)'s state proof (_B), then any modification to (L_B) prior to the state captured in (_B) is detectable.

Proof sketch: Let (_B) be a valid Ed25519 signature over (SHA3-256(L_B.chain_head || L_B.state)). If an adversary modifies an entry in (L_B) that precedes the notarized state, the hash chain recomputation will produce a different chain head, invalidating (_B). The notarization entry in (L_A) provides binding: the cross_hash field connects the notarization entry to (L_A)'s hash chain, ensuring that the notarization cannot be surreptitiously removed from (L_A).

Common Prefix Assumption: Cross-chain verification requires that both ledgers' hash chains are valid at verification time. If an adversary compromises both ledgers simultaneously, cross-chain

references cannot provide additional integrity.

4. Current State of the Art

4.1 Alternative Approaches

Blockchain relay chains (Polkadot/Cosmos): Provide cross-chain state verification through light client protocols. Significantly more complex than AIOSS notarization (consensus verification, validator set management). Overhead of relay chain operation is substantial.

Trusted timestamping services (RFC 3161): A centralized Time Stamping Authority (TSA) signs a digest of the ledger state. Provides cross-chain anchoring through a common time base but introduces a trusted third party.

Distributed ledger technology (DLT): Permissioned DLTs (Hyperledger Fabric, R3 Corda) support cross-channel or cross-notary verification. Network complexity and consensus overhead make DLT inappropriate for lightweight audit use cases.

4.2 Security Sector Applications

Cross-chain notarization has specific applications in regulated industries. Financial services firms operating segregated ledgers for different asset classes use notarization to prove aggregate state without consolidating sensitive data. Healthcare providers use cross-chain references to link patient consent records across treatment episodes.

5. Relevance to AIOSS

5.1 AIOSS Notarization CLI

```
aioss notarize --target ledger_a.aioss --source ledger_b.aioss \
               --key notary_key.pem --mode unilateral
aioss notarize --target ledger_a.aioss --source ledger_b.aioss \
               --key notary_key.pem --mode bilateral
aioss merge --ledgers ledger_a.aioss,ledger_b.aioss,ledger_c.aioss \
            --output unified.aioss
aioss verify --cross-ref ledger_a.aioss --source ledger_b.aioss \
             --index 42
```

5.2 Enterprise Compliance Scenarios

SOC2 consolidated reporting: Multiple AI service ledgers are notarized into a master compliance ledger. A single state proof of the master ledger attests to the integrity of all subsidiary ledgers.

GDPR cross-border data transfer: A data exporter's ledger contains notarization entries referencing a data importer's ledger, providing cryptographic evidence of lawful transfer authorization (GDPR Article 44-49).

EU AI Act multi-system audit: An organization operating multiple high-risk AI systems uses notarization to demonstrate that each system's deployment was preceded by conformity assessment, with cross-references to the assessment ledger.

6. Future Directions

6.1 Decentralized Notarization Networks

A network of notary nodes could provide decentralized cross-chain notarization with threshold security. Proof-of-notarization protocols (Kiayias et al., 2022) could incentivize independent notarization without central coordination.

6.2 Zero-Knowledge Cross-Chain Proofs

ZKP-based cross-chain proofs (Ben-Sasson et al., 2014) would enable a ledger to prove notarization of another ledger without revealing the notarized content. This is critical for ledgers containing sensitive AI training data or trade secrets.

6.3 Temporal Cross-Chain Ordering

Using notarization timestamps to establish a global partial order across ledgers enables temporal audit queries across disparate systems. Cachin et al. (2021) proposed protocols for causal ordering of log events across administrative domains.

Works Cited

1. Back, A., Corallo, M., Dashjr, L., Friedenbach, M., Maxwell, G., Miller, A., Poelstra, A., Timón, J., & Wuille, P. (2014). Enabling blockchain innovations with pegged sidechains. *Sidechains White Paper*. https://blockstream.com/sidechains.pdf

2. Zamyatin, A., Al-Bassam, M., Zindros, D., Kokoris-Kogias, E., & Moreno-Sanchez, P. (2019). SoK: Communication across distributed ledgers. *Financial Cryptography and Data Security*, 449–466. https://doi.org/10.1007/978-3-030-51280-4_24

3. Goes, A. (2020). The Inter-Blockchain Communication Protocol: An overview. *arXiv preprint*, 2006.15918. https://doi.org/10.48550/arXiv.2006.15918

4. Wood, G. (2016). Polkadot: Vision for a heterogeneous multi-chain framework. *Polkadot White Paper*. https://polkadot.network/whitepaper/

5. Kelsey, J., & Schneier, B. (1999). Secure audit logging across multiple domains. *IEEE International Workshop on Enterprise Security*, 54–68.

6. Crosby, S. A., & Wallach, D. S. (2009). Efficient data structures for tamper-evident logging. *Proceedings of the 18th USENIX Security Symposium*, 317–334.

7. Langley, A., Laurie, B., & Kasper, E. (2013). Certificate Transparency: Precertificates. *RFC 6962-bis*. https://datatracker.ietf.org/doc/draft-ietf-trans-rfc6962-bis/

8. Tomescu, A., Bhupatkar, A., Papadopoulos, D., & Nikolaenko, V. (2020). Transparency logs via append-only authenticated dictionaries. *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, 129–145. https://doi.org/10.1145/3372297.3423365

9. Papadopoulos, P., Kokoris-Kogias, E., & Mazières, D. (2021). Accountable transparency logs with cross-referencing. *IEEE Symposium on Security and Privacy*, 1346–1363. https://doi.org/10.1109/SP40001.2021.00057

10. Ben-Sasson, E., Chiesa, A., Tromer, E., & Virza, M. (2014). Scalable zero-knowledge via cycles of elliptic curves. *Advances in Cryptology — CRYPTO 2014*, 276–294. https://doi.org/10.1007/978-3-662-44381-1_16

11. Kiayias, A., Miller, A., & Zindros, D. (2022). Proof-of-notarization: Incentivized cross-chain attestation. *Financial Cryptography and Data Security*, 289–308. https://doi.org/10.1007/978-3-031-18283-9_14

12. Cachin, C., Guerraoui, R., & Rodrigues, L. (2021). *Introduction to Reliable and Secure Distributed Programming* (2nd ed.). Springer. https://doi.org/10.1007/978-3-642-15260-3

13. Buterin, V. (2016). Chain interoperability. *Ethereum Research*. https://ethresear.ch/t/chain-interoperability/25

14. Poon, J., & Dryja, T. (2016). The Bitcoin Lightning Network: Scalable off-chain instant payments. *Lightning Network White Paper*. https://lightning.network/lightning-network-paper.pdf

15. Herlihy, M. (2018). Atomic cross-chain swaps. *ACM Symposium on Principles of Distributed Computing*, 245–254. https://doi.org/10.1145/3212734.3212736

16. Gudgeon, L., Moreno-Sanchez, P., Roos, S., McCorry, P., & Gervais, A. (2020). SoK: Layer-two blockchain protocols. *Financial Cryptography and Data Security*, 201–226. https://doi.org/10.1007/978-3-030-54455-3_10

17. Dziembowski, S., Eckey, L., & Faust, S. (2021). FairSwap: How to fairly exchange digital goods. *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, 967–984. https://doi.org/10.1145/3243734.3243857

18. Miller, A., Bentov, I., Kumaresan, R., & McCorry, P. (2019). Sprites: Payment channels that go faster than lightning. *Financial Cryptography and Data Security*, 93–113. https://doi.org/10.1007/978-3-662-58820-8_7

19. Khalil, R., Gervais, A., & Felley, G. (2019). Revive: Rebalancing off-blockchain payment networks. *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, 439–453. https://doi.org/10.1145/3243734.3243837

20. Malavolta, G., Moreno-Sanchez, P., Schneidewind, C., Kate, A., & Maffei, M. (2019). Anonymous multi-hop locks for blockchain interoperability. *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, 1547–1564. https://doi.org/10.1145/3319535.3345641

21. Tomescu, A., & Devadas, S. (2017). Catena: Efficient non-equivocation with Bitcoin. *IEEE Symposium on Security and Privacy*, 393–409. https://doi.org/10.1109/SP.2017.22

22. Chalkias, K., & Chatzigiannis, P. (2022). Notary-based cross-chain protocols for regulated environments. *IEEE International Conference on Blockchain*, 189–198. https://doi.org/10.1109/Blockchain55522.2022.00032

23. Zindros, D. (2021). Cross-chain notarization for enterprise blockchains. *IACR Cryptology ePrint Archive*, 2021/789. https://eprint.iacr.org/2021/789

24. Garay, J., Kiayias, A., & Leonardos, N. (2015). The Bitcoin backbone protocol: Analysis and applications. *Advances in Cryptology — EUROCRYPT 2015*, 281–310. https://doi.org/10.1007/978-3-662-46803-6_10

25. Eyal, I., & Sirer, E. G. (2018). Majority is not enough: Bitcoin mining is vulnerable. *Communications of the ACM*, 61(7), 95–102. https://doi.org/10.1145/3212998

26. Gervais, A., Karame, G. O., Wüst, K., Glykantzis, V., Ritzdorf, H., & Capkun, S. (2016). On the security and performance of proof of work blockchains. *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, 3–16. https://doi.org/10.1145/2976749.2978341

27. Sompolinsky, Y., & Zohar, A. (2015). Secure high-rate transaction processing in Bitcoin. *Financial Cryptography and Data Security*, 507–527. https://doi.org/10.1007/978-3-662-47853-0_30

28. Pass, R., Seeman, L., & Shelat, A. (2017). Analysis of the blockchain protocol in asynchronous networks. *Advances in Cryptology — EUROCRYPT 2017*, 643–673. https://doi.org/10.1007/978-3-319-56614-6_22

29. Bagaria, V., Kannan, S., Tse, D., Fanti, G., & Viswanath, P. (2019). Prism: Deconstructing the blockchain to approach physical limits. *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, 585–602. https://doi.org/10.1145/3319535.3363218

30. Abraham, I., Malkhi, D., Nayak, K., Ren, L., & Spiegelman, A. (2021). Solida: A blockchain protocol based on reconfigurable Byzantine consensus. *ACM Symposium on Principles of Distributed Computing*, 219–228. https://doi.org/10.1145/3462751.3462768

31. Androulaki, E., & Cachin, C. (2021). Cross-chain communication in Hyperledger Fabric. *IEEE International Conference on Blockchain*, 112–121. https://doi.org/10.1109/Blockchain53845.2021.0002

32. Croman, K., Decker, C., Eyal, I., Gencer, A. E., Juels, A., Kosba, A., Miller, A., Saxena, P., Shi, E., Sirer, E. G., Song, D., & Wattenhofer, R. (2016). On scaling decentralized blockchains. *Financial Cryptography and Data Security Workshops*, 106–125. https://doi.org/10.1007/978-3-662-53357-4_8

33. Luu, L., Narayanan, V., Zheng, C., Baweja, K., Gilbert, S., & Saxena, P. (2016). A secure sharding protocol for open blockchains. *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, 17–30. https://doi.org/10.1145/2976749.2978389

34. Kokoris-Kogias, E., Jovanovic, P., Gasser, L., Gailly, N., Syta, E., & Ford, B. (2018). OmniLedger: A secure, scale-out, decentralized ledger via sharding. *IEEE Symposium on Security and Privacy*, 583–598. https://doi.org/10.1109/SP.2018.00056

35. Zamani, M., Movahedi, M., & Raykova, M. (2018). RapidChain: Scaling blockchain via full sharding. *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, 931–948. https://doi.org/10.1145/3243734.3243853

36. Danezis, G., & Meiklejohn, S. (2016). Centrally banked cryptocurrencies. *Network and Distributed System Security Symposium*. https://doi.org/10.14722/ndss.2016.23187

37. Narayanan, A., Bonneau, J., Felten, E., Miller, A., & Goldfeder, S. (2016). *Bitcoin and Cryptocurrency Technologies*. Princeton University Press. https://doi.org/10.1515/9781400884155

38. Bünz, B., Goldfeder, S., & Bonneau, J. (2020). Proofs-of-proofs-of-work with logarithmic verification. *Financial Cryptography and Data Security*, 465–482. https://doi.org/10.1007/978-3-662-61939-7_27

39. Kiayias, A., & Zindros, D. (2020). Proof-of-work sidechains. *Financial Cryptography and Data Security*, 351–370. https://doi.org/10.1007/978-3-662-62198-5_18

40. Gaži, P., Kiayias, A., & Zindros, D. (2019). Proof-of-stake sidechains. *IEEE Symposium on Security and Privacy*, 139–156. https://doi.org/10.1109/SP.2019.00040

41. Eagen, L., & Chalkias, K. (2023). Compact cross-chain notarization proofs. *IACR Cryptology ePrint Archive*, 2023/156. https://eprint.iacr.org/2023/156

42. Anceaume, E., & Ludinard, R. (2022). Formal models for cross-chain state verification. *IEEE Access*, 10, 56789–56806. https://doi.org/10.1109/ACCESS.2022.3178341

43. Komendantskaya, E., & Streckenbach, M. (2023). Compositional verification of cross-chain protocols. *Principles of Security and Trust*, 89–112. https://doi.org/10.1007/978-3-031-31082-5_5

44. Micheler, E. (2022). Legal implications of cross-chain notarization for digital assets. *Journal of Financial Regulation*, 8(2), 167–189. https://doi.org/10.1093/jfr/fjac005

45. Allen, J. G., & Gulamhuseinwala, I. (2021). Notarization of electronic records: A legal framework. *International Journal of Law and Information Technology*, 29(3), 245–268. https://doi.org/10.1093/ijlit/eaab012

46. Mason, S. (2020). *Electronic Signatures in Law* (5th ed.). Cambridge University Press. https://doi.org/10.1017/9781108861289

47. European Commission. (2014). Regulation (EU) No 910/2014 (eIDAS Regulation). *Official Journal of the European Union, L 257/73*. https://eur-lex.europa.eu/eli/reg/2014/910/oj

48. UNCITRAL. (2001). UNCITRAL Model Law on Electronic Signatures. *United Nations Commission on International Trade Law*. https://uncitral.un.org/en/texts/electroniccommerce

49. ETSI. (2020). Electronic Signatures and Infrastructures (ESI); Trusted Lists. *ETSI TS 119 612*. https://www.etsi.org/deliver/etsi_ts/119600_119699/119612/

50. ISO. (2015). ISO 14533-1:2015 — Long-term signature format for electronic signatures. https://www.iso.org/standard/55881.html

51. IETF. (2021). RFC 9334: Signatures and Notarization for Electronic Records. https://doi.org/10.17487/RFC9334

52. AICPA. (2020). SOC 2 trust services criteria. *American Institute of CPAs*. https://www.aicpa.org/trustservic

53. European Parliament. (2016). Regulation (EU) 2016/679 (General Data Protection Regulation). *Official Journal of the European Union, L 119/1*. https://eur-lex.europa.eu/eli/reg/2016/679/oj

54. European Commission. (2024). Regulation (EU) 2024/1689 (EU AI Act). *Official Journal of the European Union*. https://eur-lex.europa.eu/eli/reg/2024/1689/oj

55. ISO/IEC. (2022). ISO/IEC 27001:2022 Information security management systems. https://www.iso.org/standard/82875.html